**Self-introduction:**

I do have around 4.5 years of experience in IT industry as Test Analyst in Manual and automation Testing. Automation Testing exactly in to **Java selenium**. Apart from this I have good understanding in **Java and OOPS concepts**. Something related to my testing exposure, I have experience **in preparing Test plan, Test case Documen t, review and execution** and of course **bug reporting and bug tracking**. For test management I am using **Redmine and the same has been used for bug reporting too.**

I also have good exposure to Software development technologies **like V-Model and Agile**. I am part of all **Agile ceremonies (grooming, planning, standup, demo retrospect).** Currently I am associated with Legacies Media Trivandrum. In Agile we are following **SCRUM frame work**. **My roles and responsibilities** are preparing test cases, execution of test cases, bug reporting and tracking, participating in agile ceremonies etc. By education

Currently am associated with a pay roll application project for a company named ‘Caredirect Recruitment Ltd’ UK. This application has various modules such as Company, Clients, Workers, Deductions, Payslip, Report etc. There is an admin, who can create company, clients and workers. Mapping between company, clients and workers are also done. Users are given credentials for login so that they can update their daily work time schedules and apply for leaves. There is also option for adding deductions based on rent, insurance etc. so that final payslip can be generated easily with all the details like working hours, deduction etc.

I am M.Tech Graduate in Telecommunication Engineering from Kerala University. **I think am done with me.**

**Explain blackbox testing?**

In black box testing is done without knowing the code, algorithm, internal structure knowledge. It is done by the testers by preparing test cases based on test case techniques and applying them on Acceptance criteria for a particular user story.

Blackbox testing methodologies: Boundary value analysis, Equivalence class partitioning, Decision Table, State Transition table, Error testing, Experience based testing.

**Explain White box testing?**

This type of testing is done by having knowledge about code or algorithm. It is done by developers. Junit test cases can be used for unit testing cases. This is can be verifying by using http request and response.

**How user stories are prepared?**

From use case diagram Business Requirement Document is created, which is used for creating Software Requirement Document. From this document, features are groomed to create user stories.

**Explain Regression Testing:**

Whenever a new set of code changes in a build, we need to make sure that the change in code will not affect the exiting functionalities of the build.

Egs: changes in the settings page,

If there is a change in employer module su,ch that new form accept DoB and photo. Then changes will occur in employee details page, Report page, My profile page. Then we need to go for regression testing.

**Explain Sanity Testing:**

It checks whether the build is ready to move to another environment. It is also a subset of regression testing. It also checks whether the build is ready for further release. It may also be considered as last round of confirmation testing.

Explain Smoke Testing:

It checks whether the build is fit for further testing. It also checks the high level functionalities of the build.

**Explain CRM?**

Prepared BRS will be submitted to client and client will give necessary review comments and suggestions. The review comments and suggestions are incorporated and submitted for further approval. This process is called Change Request Management and the out of this process is known as Baseline requirements.

**When we are expected to start testing?**

* Baseline requirements are finalized
* Test cases are reviewed and approved
* User story has resolved status
* Test data is available and Test environment is setup

**Explain status of User story?**

New

Active

Resolved

Closed

**Explain Resolved status of user story?**

All dev task should be completed

All code review should be completed

Unit Testing should be completed.

**Explain Bug life cycle?**

Bug is logged when the expected results and actual result is not met

Bug is raised by the tester and reported into the Test Management Tool and set the status as New

Developer will validate the bug by using steps to reproduce and makes the status as active/open if valid. If bug is invalid he may assign any of the following status based on the scenario.

* He may change New to ‘Duplicate’ if the bug is already opened in the Test Management Tool
* He may change New to ‘Deffered’ if the bug is not fixed now and may be fixed in next sprint. (For the deffered bug Tester has to get confirmation from BA)
* He may change New to ‘invalid’ if there is conflict of understanding
* He may change New to ‘cannot be reproduce’ if fails to reproduce the bug

If found valid, then developer make the status act as ‘Active’. If he started working on the bug, he will change the status as ’In progress’. If he has resolved the bug, he will set the status as Ready for retest/Fixed and assigned back to the Tester,

Tester will retest and make status as closed if expected conditions are met. If failed, then it will be reassigned back to the dev with status as New.

**Explain TestCase and Test Scenario?**

A Test Scenario is a statement describing the functionality of the application to be tested. It is generally considered as “what is to be tested”?. A single test scenario can cover one or more test cases. Therefore, a test scenario has a one-to-many relationship with the test cases. Test case is step by step procedure to execute testing. It is generally considered as “how to be tested”?.

**What is Browser compatibility testing and UI testing?**

UI testing is a functionality testing. It checks alignments, css styles, Images and templates

Browser compatibility testing is a type of UI testing checks the behavior of the functionality in different browsers, devices, or in desktop configurations. It checks file uploading, session handling, fonts, css styles, Image rendering, document upload functionalities. It also checks proper loading and retrieving of images, css styles etc.

**Explain the testcase for logout?**

Verify that Logout button is visible or not.

Verify that login page appears after clicking logout button.

Verify that user is able to login after clicking log out button.

Verify that user is logged with the same URL even if the logout page is closed.

Verify that user is logged in even if the back button is clicked

Verify that user is logged out after a specifc time.

Verify that user is able to login from different browsers and devices

Verify that user is able to login with different username

Verify whether logs contain same IP for the same ID.

**Explain the test case for Registration Page?**

Verify that registration page has following fields (a) First Name (b)Middle Name (c) Last Name (d)Address (e) Zip Code (f) Image Upload (g) Company

Verify that submit button is visible.

Verify that mandatory fields are marked as “\*”

Verify that First Name does not take characters

Verify that validation message appears for the mentioned fields

Verify that user is not able to submit the page with blank details

Verify that Registration button is visible

Verify that clicking the registration tab loads the registration page

Verify that upon submit values should be be validated with database.

Verify that “Form submitted successfully” message is displayed if the form is submitted by filling the form with mandatory fields.

Verify that “Please fill the mandatory details” message is displayed if the form is submitted without filling mandatory fields.

Verify the maximum and minimum fields of each fields

Verify that error message is displayed if the image is not as per the specification.

**Describe about Test case for internet login.**

Verify that the login screen contains elements such as Username, Password, Sign in button, Remember password check box, Forgot password link, and Create an account link.

Verify that cursor is focused on “Username” text box on the page load (login page)

Verify that all the fields such as Username, Password has a valid placeholder

Verify that User is able to Login with Valid Credentials

Verify that User is not able to Login with invalid Username and invalid Password

Verify that User is not able to Login with Valid Username and invalid Password

Verify that User is not able to Login with invalid Username and Valid Password

Verify that User is not able to Login with blank Username or Password

Verify that User is not able to Login with inactive credentials

Verify that clicking on browser back button after successful login should not take User to log out mode

Verify that clicking on browser back button after successful logout should not take User to logged in mode

Verify that there is a limit on the total number of unsuccessful login attempts

Verify that the ways to retrieve the password if the User forgets the password

Verify that “Remember password” checkbox is unselected by default

Verify that “Keep me logged in” checkbox is unselected by default

Verify that the timeout of the login session (Session Timeout)

Verify that the logout link is redirected to login/home page

Verify that User is redirected to appropriate page after successful login

Verify that User is redirected to Forgot password page when clicking on Forgot Password link

Verify that User is redirected to Create an account page when clicking on Sign up / Create an account link

Verify that validation message is displayed in case when User leaves Username or Password as blank

**Explain Test Plan?**

It is prepared by senior tester. It is a low level document. It is also called as root map of testing. It has following sections:

* Scope and objective
* Features to be tested
* Test Configuration
* Schedule and Deliverables
* Test Management Tool
* Resources
* Test Environment
* Test Deliverables
* Test Approach
* QA signoff

**What are the severity and priority levels of bug?**

Severity gives the impact of bug in the application, while priority gives the order in which bugs are to be fixed

Severity levels: Critical, blocked, Major, Cosmetic

Priority levels: Low, Medium and High

**Explain BVA and EP?**

**Boundary Value Analysis and Equivalence Partitioning** both are test case design strategies in Black-Box Testing. **Black box testing**, also known as Behavioral Testing, is a [software testing method](http://softwaretestingfundamentals.com/software-testing-methods/) in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

**Boundary Value Analysis:** It's widely recognized that **input values at the extreme ends of the input domain cause more errors in the system**. More application **errors occur at the boundaries**of the input domain. ‘Boundary Value Analysis' Testing technique is used to identify errors at boundaries rather than finding those that exist in the center of the input domain.

### Equivalence Partitioning: In this method, the input domain data is divided into different equivalence data classes. This method is typically used ****to reduce the total number of test case****s to a finite set of testable test cases, still covering maximum requirements.

### Describe Test Cases for Profile Picture uploading?

**Test cases for Profile picture uploading.**

* Verify that there is browse option to search picture from the device.
* Verify there is option to take picture.
* Verify that the selected picture should be of correct format, ie JPG or PNG.
* Verify that size of the picture should not be less than 50kB and should exceed 10MB.
* Verify that there is option to adjust the picture to the given space.
* Verify that pixel size should be of correct format.
* Verify the time taken to Upload the picture.
* Verify that a progress bar should be displayed if uploading takes more than 10 sec.
* Verify there is an option to save the picture.

**RTM (Requirement Traceability Matrix)**

Feature of test management tool which link requirement to test cases. It is a bidirectional Traceability matrix.

In test management tool epic is linked with feature, which is linked with user story, which is linked with test cases. It also has the status of test case execution, Passed test cases, failed test cases, pending test cases and bug. Therefor by looking into RTM we will get the full details for works.

Some common questions:-

* A release is set of 10th and you are asked for the status report on 5th where will you get the status.
* A release is set of 10th and how will you make sure that you are ready for release and make sure you have done enough testing.
* Check all the requirement is linked to test case and all the user story are linked with test cases.
* What is the status of todays test execution. Where will you extract the report.

**When to close test execution or ready to release.**

* All test cases should be completed.
* There should be no failed test cases.
* There should no high/critical active/open bug.
* Release date should be met.
* User story status should be closed.

**What is retesting?**

**Retesting**

Testing the previously failed test cases again to verify whether the defects posted earlier are fixed or not.

Differentiate Error, Defect, Failure and Bug?

Error: Unhandled exception thrown by a line of code in Dev environment

Defect: If the error in QA raises again in Prepod or UAT, then it called as defect. It can be recorded in Customer Relationship Management Tool or in excel

Failure: Issues raised by the customers in live production environment and product is called faulty product.

**Explain levels of testing?**

Unit Testing: It is usually done by developers for each of the units. Units consists of UI, Services, Database. Testers do unit testing using Junit.

Integration Testing: It is called as user story testing. Different components are integrated and tested the interfaces

System Testing: Different user stories are tested. System is considered as a single unit and end to end testing is done.

User Acceptance Testing:

Alpha Testing: It is done to check whether to accept or reject the build before releasing the build for the customers for UAT. It is also called as internal acceptance testing and performed in Dev environment by Tester using end user mind.

Beta Testing: It is done in live or client environment and build that is going to test is called as beta version. It is eventually limited to limited number of audiences and changes are done based upon their feedback.

**Write Negative Test Cases for ATM**

ATM accepts the card even if invalid PIN is entered

Amount entered into the machine and dispatched is different

**Write Negative Test Cases for TextBoxes**

Typed data is not displayed in textbox

**Explain deffered, duplicate and Invalid/rejected status in bug?**

Deffered: It is done by developer. Then tester need to clarify this with the BA. Based on BA’s input further decision is made.

Invalid/Rejected: This happens when developer understanding of AC of user story is different from Tester. In this case contact with BA for further decision.

Duplicate: This is initiated by developer. If raised Tester look at Test management Tool and track the bug, if it is already in opened status, then make raised new bug as closed.

Explain Peer-Peer Review?

It is basically coworker review. Here somebody in the same designation reviews it. If there occurs conflict then it will be resolved by BA, Test Lead.

What is Test data?

Set of input values or data that we are giving to application interface at the time of testing.

Test cases for elevator?

* Verify the height and width and volume of life as per the requirement.
* Verify that button for closing lift, opening lift, fan, emergency, and all floor numbers should be there on the button panel
* Verify the presence of display where floor number appears
* Verify floor number is being announced on each floor
* Verify that the maximum number of people and weight is written on the wall and also verify all emergency instructions.
* Verify that lift is able to move up and down
* Verify that lift is capable to stop on floors which are pressed in the button panel
* Verify that lift close when the close button is placed or should close once after few time
* Verify that fan is working on pressing the respective button
* Verify the maximum weighing capacity of life by putting weight
* Verify that light and fan is running at power failure
* Verify that lift stops at right floor i.e if user click on 2nd floor then lifts should stop on 2nd floor, not on 4thfloor.
* Verify the landing speed of the lift
* Verify the time between two successive floors.
* Verify that lift doors can’t be open when the lift is moving down or upward
* Verify that lift doors are closed when no one is using it
* Verify that once lift has crossed the certain floor and some person has to click to open lift then lift should open when lift comes down at that floor.
* Verify working of lift moves to very next floor on power failure
* Verify the working of alarm in lift once weight exceeds the standard weight.

**Our final product release today at 4pm, tester got a critical bug at 3:30pm. How to handle this situation?**

Sent an email to the client stating that release criteria is niot met and hence QA signoff cannot be given.

**After delivering a product, client reported a major issue in Login Page remainder section. How to handle this situation and wrtite down test cases:?**

Check the bug in preprod and dev, if arises fix it immediately in UAT environment, and this type of fixing bug is called as hot fix/patch release. If bug is not raised in dev environment, then it may be a configuration issue. This is done by skipping certain dev process. RCA such as fishborn diagram, Ishikawa diagram, 5Y etc is done to evaluate this. Check whether test cases are available, if available check whether it is pass/fail. Also check all the test cases are executed or not etc.

**I got 10 test cases, 5 test cases passed and remaining 5 is failed. How many test cases will be performed for Regression testing and Retesting in remaining 5 failed test cases.**

As per theory: for regression 10 test cases. In practical cases: for regression 5 test cases, for retesting 5 test cases.

**Our release on tomorrow, I have 1000 test casesinthecurrentproject.Howmany test cases will be tested for the particular release?**

Select priority test cases P1, P2 from Test Management Tool into a test suite. Based upon the available time execute the testcases and publish signoff letter.

**I have prepared 1000 test cases for my current project, how many test cases tested for Sanity and Smoke testing.**

Basic functionalities are tested in Smoke Testing while in sanity critical functionalities are checked. Generally as per the application 800 testcases for sanity and 100 test cases for smoke testing.

**Simon is a Tester, he got a new web application for testing. He doesn't know any functionality or scenario of the application. How to test this application?**

Do black box testing by using test case techniques to prepare test cases. Exploratory testing is done with domain and without product knowledge. It may or may not have test cases. Adhoc testing can also be done, here adhoc scenarios along with product knowledge is required and document has to be prepared for the test execution.

**Explain Drag and Drop in selenium**

//WebElement on which drag and drop operation needs to be performed

WebElement from = driver.findElement(By.id("sourceImage"));

//WebElement to which the above object is dropped

WebElement to = driver.findElement(By.id("targetDiv"));

//Creating object of Actions class to build composite actions

Actions act = new Actions(driver);

//Performing the drag and drop action

act.dragAndDrop(from,to).build().perform();

**Right click action in selenium Scroll Down**

**a).** Actions action = new Actions(driver);

action.moveToElement(element).click().perform();

**b).** Actions actions = new Actions(driver);

WebElement elementLocator = driver.findElement(By.id("ID"));

actions.contextClick(elementLocator).perform();

**Note: Mouse actions in Action class.**

click(): Clicks at the current mouse location.

doubleClick(): Performs a double-click at the current mouse location.

contextClick() : Performs a context-click at middle of the given element.

clickAndHold(): Clicks (without releasing) in the middle of the given element.

dragAndDrop(source, target): Click-and-hold at the location of the source element, moves to the location of the target element

dragAndDropBy(source, xOffset, yOffset): Click-and-hold at the location of the source element, moves by a given offset

moveByOffset(x-offset, y-offset): Moves the mouse from its current position (or 0,0) by the given offset

moveToElement(toElement): Moves the mouse to the middle of the element

release(): Releases the depressed left mouse button at the current mouse location

**How to take a Screenshot in selenium**

**public** **void** tearDown (ITestResult result)

{

**if** (ITestResult.***FAILURE*** == result.getStatus()) {

**try** {

TakesScreenshot ts = (TakesScreenshot) driver;

File source = ts.getScreenshotAs(OutputType.***FILE***);

//Move image file to new destination

File DestFile=**new** File("./target/" + result.getName() + ".png");

//Copy file at destination

FileUtils.*copyFile*(source, DestFile);

//GenericUtility.copyFileUsingStream(source, new File("./target + result.getName() + ".png"));

System.***out***.println("Screenshot taken");

} **catch** (Exception e) {

System.***out***.println("Exception while taking screenshot " + e.getMessage());

}

}

}

**How to Get Current page url in selenium**

*String url = driver.getCurrentUrl();*

**Count the character length of current url**

driver.getCurrentUrl().length()

**How to Handle window in selenium**

ArrayList<String> tab = **new** ArrayList<String>(*driver*.getWindowHandles());

System.***out***.println("Windows values " + tab);

*driver*.switchTo().window(tab.get(2));

*driver*.get("https://www.facebook.com");

*driver*.switchTo().window(tab.get(0)); // switch back to main screen

*driver*.get("https://www.news.google.com");

Actions builderr = **new** Actions(*driver*);

String title = "Facebook – log in or sign up";

**for** (String handle : tab) {

System.***out***.println("Windows Title is " + *driver*.switchTo().window(handle).getTitle());

**if** (*driver*.switchTo().window(handle).getTitle().equals(title)) {

//System.out.println("jhjhjhj");

**break**;

}

}

}

**Switch to pop up**

1) void dismiss() **// To click on the 'Cancel' button of the alert.**

driver.switchTo().alert().dismiss();

2) void accept() **// To click on the 'OK' button of the alert.**

driver.switchTo().alert().accept();

3) String getText**() // To capture the alert message.**

driver.switchTo().alert().getText();

4) void sendKeys(String stringToSend) **// To send some data to alert box.**

driver.switchTo().alert().sendKeys("Text");

**How to Identify a Text box**

**String typedText = driver.findElement(By.xpath("xpath\_textbox")).getAttribute("type"));**

Webelement.isdisplayed or isenabled.

**How to Switch to tab**

ArrayList<String> tab = **new** ArrayList<String>(*driver*.getWindowHandles());

System.***out***.println("Windows values " + tab);

*driver*.switchTo().window(tab.get(2));

*driver*.get("https://www.facebook.com");

*driver*.switchTo().window(tab.get(0)); // switch back to main screen

*driver*.get("https://www.news.google.com");

**What is the difference between Close and Quit**

The driver.close() command is used to close the current browser window having focus. In case there is only one browser open then calling driver.close() quits the whole browser session.

Usability  
It is best to use driver.close() when we are dealing with multiple browser tabs or windows e.g. when we click on a link that opens another tab. In this case after performing required action in the new tab, if we want to close the tab we can call the driver.close() method.

|  |  |
| --- | --- |
|  | //Closing the tab  driver.close(); |

## driver.quit()

The driver.quit() is used to quit the whole browser session along with all the associated browser windows, tabs and pop-ups.

**Usability**  
It is best to use driver.quit() when we no longer want to interact with the driver object along with any associated window, tab or pop-up. Generally, it is one of the last statements of the automation scripts. In case, we are working with Selenium with TestNG or JUnit, we call driver.quit() in the @AfterSuite or method . Thus, closing it at the end of the whole suite.

|  |  |
| --- | --- |
|  | @AfterSuite  public void tearDown() {     driver.quit();  } |

**How to launch a url**

driver.get("http://www.google.com");

driver.navigate().to("http://www.google.com ");

Note:

1. get() method does not allow driver to access the browser’s history while navigate() allows the driver to access the browser’s history and to navigate to a given URL.
2. You can not navigate back and forward using get() while same can be achieved through navigate().
3. There is no specific method for refresh a web page in get() while navigate() provides a method named refresh(). You can refresh a page using get() also by calling get() method again.

**TestNG annotations and Order of annotations**

**TestNG Annotations:**

**@Test:** To mark a method as a test method.

**@BeforeMethod:** performed before each test (@Test) method.

**@AfterMethod:** executed after each test method.

**@BeforeClass:** Performed before the first examination method in the current class.

**@AfterClass:** Executed after all test methods in the current class.

**@BeforeTest:** performed in this suite before any test methods of classes available inside <test> tag in testng.xml.

**@AfterTest:** Execute this room in testng.xml after all test methods available in the class <test> inside the tag.

**@BeforeSuite:** Performed before all tests in this suite.

**@AfterSuite:** Executed after all tests executed in this current suite.

**@BeforeGroup:** In the specified group, the first test method is executed before.

**@AfterGroup:** Executed after the end of all test methods executed in that specific group.

* There are more annotations in TestNG @BeforeGroup, @AfterGroup, @Listeners, @Parameters, @DataProvider, @Factory.

**Execution Sequence of TestNG Annotations:**

1. BeforeSuite
2. BeforeTest
3. BeforeClass
4. BeforeMethod
5. Test
6. AfterMethod
7. AfterClass
8. AfterTest
9. AfterSuite

**Data read from excel**

**public** **static** String getCellValue(String xl, String Sheet, **int** r, **int** c)

{

**try**

{

FileInputStream fis = **new** FileInputStream(xl);

Workbook wb = WorkbookFactory.*create*(fis);

Cell cell = wb.getSheet(Sheet).getRow(r).getCell(c);

**return** cell.getStringCellValue();

}

**catch** (Exception e)

{

**return** "Error in loading data!!";

}

}

**How to generate a test report in selenium and what is commonly we are using method**

### Generate Reports Using TestNG

Report is find in the <***index.html***> file. It combines detailed information like the errors, test groups, execution time, step-by-step logs and TestNG XML file.

It is the trimmed version and informs about the no. of “Passed”/”Failed”/”Skipped” cases. You can see it from the <***emailable-report.html***> file. It’s an email-friendly report which you can embed and share with the stakeholders.

**What is the difference between Find Elements and Find Element**

|  |  |
| --- | --- |
| **Find Element** | **Find Elements** |
| Returns the first most web element if there are multiple web elements found with the same locator | Returns a list of web elements |
| Throws exception NoSuchElementException if there are no elements matching the locator strategy | Returns an empty list if there are no web elements matching the locator strategy |
| It will only find one web element | It will find a collection of elements whose match the locator strategy. |
| Not Applicable | Each Web element is indexed with a number starting from 0 just like an array |

**TestNG** is an automation testing framework in which NG stands for "Next Generation". Using TestNG, we can generate a proper report, and you can easily come to know how many test cases are passed, failed, and skipped. We can also execute the failed test cases separately.

major advantages of TestNG are :

It gives the ability to produce HTML Reports of execution

Test cases can be Grouped & Prioritized more easily

Parallel testing is possible

Generates Logs

Data Parameterization is possible

**What is polymorphism in Java**

Combination of overloading and overriding is known as Polymorphism. We will see both overloading and overriding below.

Polymorphism allows us to perform a task in multiple ways.

There are two types of Polymorphism in Java:

1. Compile time polymorphism (Static binding) – Method overloading

A class having multiple methods with same name but different parameters is called Method Overloading. At compile time, Java knows which method to invoke by checking the method signatures. So this is called compile time polymorphism or static binding.

Eg: We use implicit wait in Selenium. Implicit wait is an example of overloading. In Implicit wait we use different time stamps such as SECONDS, MINUTES, and HOURS etc.

1. Runtime polymorphism (Dynamic binding) – Method overriding

Declaring a method in child class which is already present in the parent class is called Method Overriding.

In simple words, overriding means to override the functionality of an existing method.

In this case, if we call the method with child class object, then the child class method is called. To call the parent class method we have to use **super** keyword.

Eg: Examples are get and navigate methods of different drivers in Selenium.

*Why we call it as runtime polymorphism: When a parent class reference refers to the child class object then the call to the overridden method is determined at the runtime. So it is called runtime polymorphism. It is because during method call which method (parent class or child class) is to be executed is determined by the type of an object.*

**What is Inheritance in Java**

We create a Base Class in the Framework to initialize WebDriver interface, WebDriver waits, Property files, Excels, etc., in the Base Class.

We extend the Base Class in other classes such as Tests and Utility Class. Extending one class into other class is known as Inheritance.

In inheritance, we use two keywords namely extends and implements

**What is Interface in Java**

Basic statement we all know in Selenium is WebDriver driver = new ChromeDriver();

WebDriver itself is an Interface. So based on the above statement WebDriver driver = new ChromeDriver ();

we are initializing ChromeDriver browser using Selenium WebDriver. It means we are creating a reference variable (driver) of the interface (WebDriver) and creating an Object. Here WebDriver is an Interface as mentioned earlier and ChromeDriver is a class.

An interface in Java looks similar to a class but both the interface and class are two different concepts. An interface can have methods and variables just like the class but the methods declared in interface are by default abstract. We can achieve 100% abstraction and multiple inheritances in Java with Interface..

Earlier we have learnt abstract class. We learnt that with abstract class we can achieve partial abstraction and with interface we can achieve 100% abstraction. Let’s see how we can achieve 100% abstraction with Interface .

An interface in Java looks similar to a class but both the interface and class are two different concepts. An interface can have methods and variables just like the class but the methods declared in interface are by default abstract. We can achieve 100% abstraction and multiple inheritance in Java with Interface.

1. Java interface represents is-a relationship similar to inheritance

2. Interface cannot be instantiated same like abstract class

3. Java compiler adds public and abstract keywords before the interface methods

4. Java compiler adds public, static and final keywords before data members

5. Interface extends another interface just like a class extends another class but a class implements an interface.

6. The class that implements interface must implement all the methods of that interface.

**What is Multiple Inheritance in java**

Multiple Inheritance is a feature of object oriented concept, where a class can inherit properties of more than one parent class.

interface Printable{

void print(String name);

}

interface Showable{

void show(int roll);

}

class A **implements** Printable,Showable{

public void print(String name ){System.out.println("Name is "+name);}

public void show(int roll){System.out.println("Roll No is "+ roll);}

public static void main(String args[]){

A obj = new A();

obj.print("TOM");

obj.show(12);

}

}

**Notes on Agile**

Word meaning of agile – Flexible, it is code to changes

Methodology –Agile

Different frameworks in Agile – Scrum, kanban, scrumban, extreme programming, Scaled Agile( Safe Agile), Nest Scrum

In Scrum- more Customer satisfaction and project success

In scrum, working demo is reviewed in each sprint.So if changes needed , those changes can be implemented.

But in waterfall model at the end of project the clients reviewing and if any changes needed, it is difficult. Since the product completed according to requiremnts said by clients and at end of product review if changes is said it consume extra cost. So it will be great loss (in terms of money and time) so to overcome this difficulty agile concepts are used. In agile, at any time we will be able to change or modify the product.

Waterfall model fail is in the last moment

But in case of Agile fail in any moment of life span

That is early feedback and early fail is basic of agile

In agile, phase by phase demo is shown to clients and getting feedback from clients. And at regular interval developing the product in small increments and delivering to clients. So at the end of the day by known or unknown client will be a part of the process. So this way active involvement is only possible in agile and leading framework in agile is scrum.

Scrum founders are –

Firstly scrum was single entity or single company then later it splited into scrum alliance and scrum.org

We are discussing scrum.org

The 12 agile principles of scrum.org

1. Satisfy the customer

In each phase of the product development when customer is also involved in development and getting customer’s feedback in each phase and upgrading the product will make the client happy. Otherwise at end of product development the customer will be unhappy.

The customer is involved in development process. So customer interaction is more in agile compared to other models so customer will have good satisfaction. Customer interaction occurs mainly at time of sprint review i.e. interaction with scrum team is possible at time of demo or sprint review.

2. Welcome change

In a product development phase it is better to make changes at any moment rather than making changes at the end of product development.

So in agile changes can be proposed at any moment. Agile can welcome changes at any moment of the product life span.

3. Deliver frequently

Depending on sprint cycle the product is delivered frequently. Frequently means the product modules can be delivered to clients twice in a month depending on sprint so that reviewing and making changes is possible with good satisfaction. [Facebook sprint duration is single day.]

4. Work together

In older times, developers will report to development lead, tester will report to testing lead, project managers will report to their corresponding office. The same team working for a particular product reports to different leads. Then the corresponding teams work based on their managers choices and reviews so there will be conflict between all peoples. But in scrum developers, testers, and all people’s reports to same person and on work base they report to scrum master. Work related duties are handled by scrum master. For any facility management issue or people issue one line manager will be there. So entire team will be working as a single team. So can clear doubts and coordinate easily.

5. Trust and support

If out of 10 test cases 5 test cases delivered and other 5 not delivered then responsibility is for scrum team.

Example - In two week sprint cycle -10 days of working days, if release is on last Friday. The developer completes developing and given for testing on Thursday. Then tester will get time or will not get time to complete testing depending on the complexity of the test cases that are executed. Major responsibility is for concern developer and tester. And whole responsibility on scrum team. The delivering of whole user story depends on the combined effort of concerned developer and tester. So both of them adjust and have to work according to that. So for delivering the user story on time it is the responsibility, mutual trust and support of the team.

6. Face to face Conversation

In scrum all the members in the team are working together, so can ask the doubts or about bugs to the members face to face than asking by sending mail. That is more effective.

7. Working software

In agile the prior important should be given to working software rather than process and procedures.

If release is within 6 months then there is enough time to prepare the test plan document. But if release is within two months there will not be enough time to prepare the test plan document. So even if test plan document is not prepared also the release will go ahead. That means the priority is given to working software than the process and procedures. It’s not said as no need of documents but priority is less for them while comparing with working software.

8. Sustainable Development

It’s better to deliver a product phase by phase as increment. In phase by phase development process there should be constant pace between each phase i.e sustainable development.

9. Continuous attention

Product is delivered phase by phase. So in each phase one feature is delivering so it get its attention or importance and gradually integrating the developed phase to the base product. At each time of integration, regression test is carried out so a continuous testing is done on the product. So continuous attention on the product.

10. Maintain simplicity

In each phase taking independent module and developing that module and integrating to base product. So complexity less i.e. simplicity is maintained

11. Self-organizing teams

In scrum meeting status is discussed and doubts are clarified, assigning the works are done. In scrum team the members interacts within that team and self-organize.

12. Reflect and adjust

At time of each demo or review the changes are incorporated to the product i.e. reflect and adjust. The changes are reflected to product and adjust according to that.

Scrum Ceremonies

1. Scrum Grooming

Grooming – Preparing

Sprint Grooming means grooming the needs of a Sprint

Sprint needs- User Story

• Sprint Grooming means creation of user story from the respective features.

Grooming output is well groomed user stories.

How to groom user story? –

Entire team along with BA analyse the user story in scrum grooming.

Identified Features are sliced into independent user story

In grooming the user story - Summary, DOD, AC are prepared in grooming

Story point is prepared in scrum planning

Groomed user story are moved to sprint backlog (a folder)

Backlog – a collection of items

The collection of user stories that we are prepared to be taken up in the upcoming sprint those collection of user story will be kept in a separate folder in the test management tool. That particular folder that contains couple of user story that may or may not be part of the next sprint. Those collection of user story lies in specific folder can be called as Backlog.

Sprint Grooming output is collection of user story

When you are expected to perform grooming? - Before starting of a sprint.

2. Scrum Planning

The process of selecting user story from the collection of user story from grooming (backlog) to the current sprint is called as Sprint Planning.

Planning- How to do, what to do, when to do, who to do

In sprint planning -

Which all user story have to be done, when those user story get completed which all features will be completing, if so who all will be doing those features, what are the task of that user story, have to identify how much work can be done or taken by the team on that sprint…

Steps in planning- go to backlog and take first story and Ask to team how much story point to this user story

Story point- how much complexity for that task

Planning can be done in two ways : Capacity planning and Story point estimation.

Capacity planning – Planning with help of available capacity of the user story is called as capacity planning

**Explain story point estimation?**

Sprint planning done with the help of story points of user stories is called story point estimation.

During planning session, first user story is taken from the backlog and set us current sprint backlog. Planning is done at the first day of sprint and if the duration is 2 weeks, then we are expected to spend 4 hrs for sprint planning. If the sprint is 4 weeks, then the sprint planning duration is expected to 8 hours.

**When will be story point estimation staryed?**

Story point estimation need historical data. So capacity planning have to be used for 3 or 4 sprint

What is Spike user story?.

It can be called as Spike user story/research user story- analyse/ research user story.

Sometimes there will be clarification needed in domain s like whether feasible or not, depending upon the requirements said by clients. In that case those user story is taken as research or analyse story. Based on this analysis the user story is taken for development work in next sprint.

Note: Grooming can be continuous process in the middle of the sprint. At middle of a sprint, the grooming for next sprint will be carried out.

**Discuss about Sprint demo or review?**

Sprint demo or review is set to second last day of sprint so that clients are given liberty for updation. In that case analysis is done to check whether the new change can be completed in this sprint, if yes then changes are done else a new user story is created with this updation and set as high priority so that this change can be taken first in next sprint

**How demo is shown to client?** Through skype or Microsoft teams or go to meeting or zoom

Discuss about Daily Stand-up or Scrum Meeting ?

Daily stand up meeting will be started from second day of sprint and Scrum master holds the meeting. Duration is 15mts. It will be conducted from second day of sprint to last day of sprint. In scrum meeting following things are discussed:

* 1. What you have done yesterday
  2. what you are planning for today

iii. Do you have any impediments. If any impediments scrum master will solve.

Explain Retrospective meeting?

Sprint Retrospective is conducted on the last day of the sprint. It is a 15-20 mts meeting. Scrum master will be holding the meeting. Following things will be evaluated

1.What went well,

2.what didn’t went well,

3.how can we improve/lessons learned

Explain need of agile technology?

If we are using agile, we have the privilege or liberty to change the product strategy or product requirement at any point of development. Moreover if we have a dynamic project or dynamic requirement, then the best approach is agile .

**Compare agile and waterfall methodology Water fall model**   
**waterfall**

1.Requirement is fixed

2. Unidirectional (top down approach) (from testing we cannot go back to development)

3. Only at the end of the development phase the client is able to view the product and changes is difficult.

**Agile**

1. Requirements are not fixed

2. Bi directional

3.Good client interaction.

4. Infrequent interval client has liberty to see the product.

5. Changes can be made at any level or phase of development

**What is spillover?**

* Spillover Spillover is a backlog item that has failed to meet the criteria defined in the Definition of Done (DoD) for the project team.
* Spillover happens if:
  + Change in requirements after the starting of sprint.
  + Clients requirements change at the intermediate stage.
  + If any planning goes wrong (if there is third party working in that project, there may a chance of failing)

**Advantages of Waterfall model**

* Used in short term project
* Clarity in requirements
* Client should have a better understanding of product and have faith in that development team (experienced dev team)

**Sprint duration?**

Duration of sprint Monthly sprint is followed-21 days. First 15 days, new userstories testing will be conducted, during 4 days, regression testing will be conducted and 2 days will be reserved for UAT testing. In the last day, release of the product will be conducted.

**Give challenges in testing?**

* If the checkboxes in a website is dynamically changing, automation is difficult
* At some case, cannot switch to the pop up because there is a frame. So first switch to that frame and done the operations, then unselect the frame.
* Report validation, image comparison is difficult in automation

**How to automate captcha?**

Captcha is a third party plugin, so need to test the Captcha in any application. Testers will disable the Captcha in the database

**Is 100% automation possible?**

* It depends on the test cases, and report validation, image comparison is difficult in automation
* If the checkboxes in a website is dynamically changing, automation is difficult